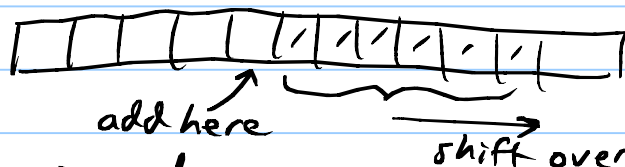


# Pointer-based Structures

Note Title

10/23/2007

## Arrays

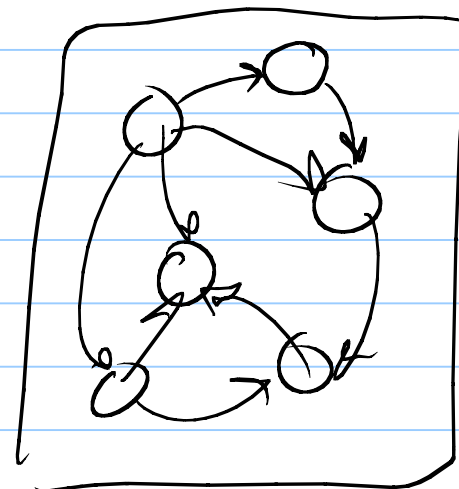


contiguous  
section of memory

- know size in advance
- random access into the collection (by index)
- regular structure — organized in a line

## Pointer-based structures

- allocate little by little
- spread over the heap
- free-form structure  
(not "boxy")
- splice in pieces w/o moving  
existing pieces



Pointer-based list of ints



```
class ListItem {  
    int value;  
    ListItem next;
```

```
    public ListItem(int v, ListItem next) {  
        value = v;  
        this.next = next;  
    }
```

```
}
```

```
class ListItem {  
    int value;  
    ListItem next;
```

```
public ListItem(int v, ListItem next) {  
    value = v;  
    this.next = next;  
}
```

ListItem a, b;

a = new ListItem(5, null);

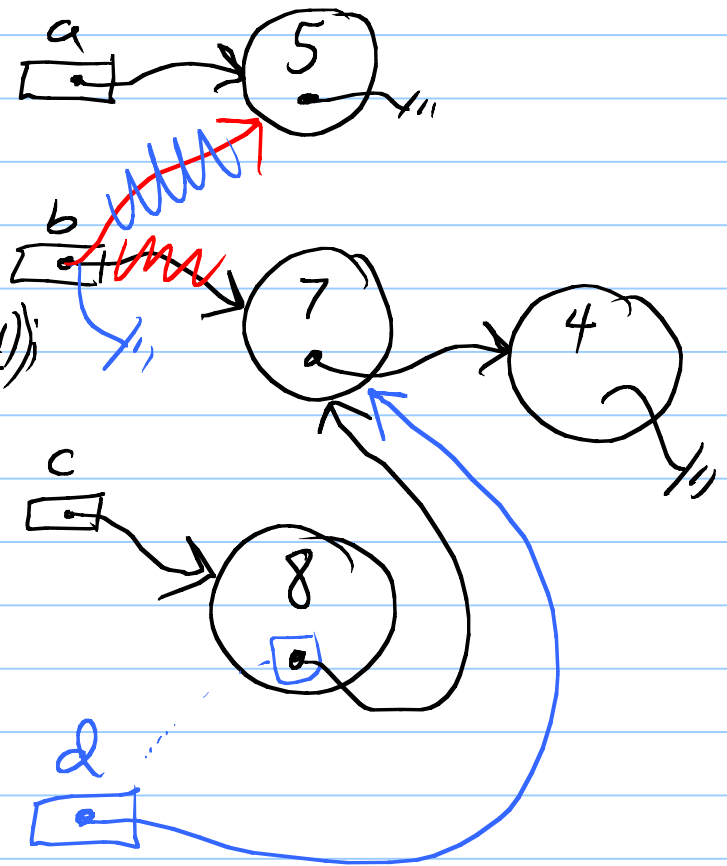
b = new ListItem(7, new ListItem(4, null));

ListItem c = new ListItem(8, b);

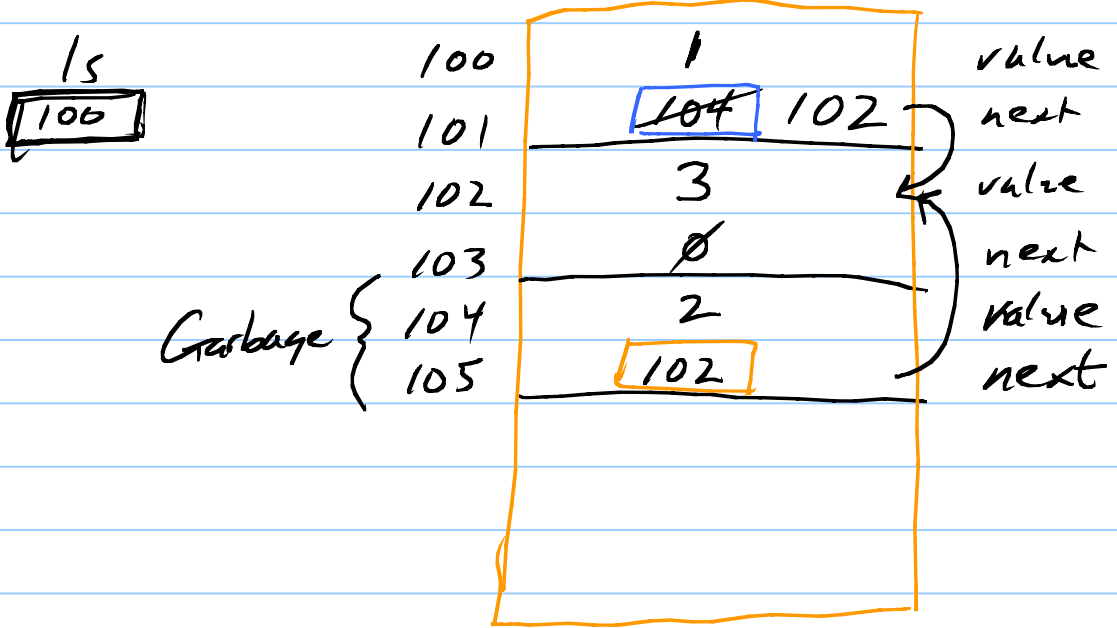
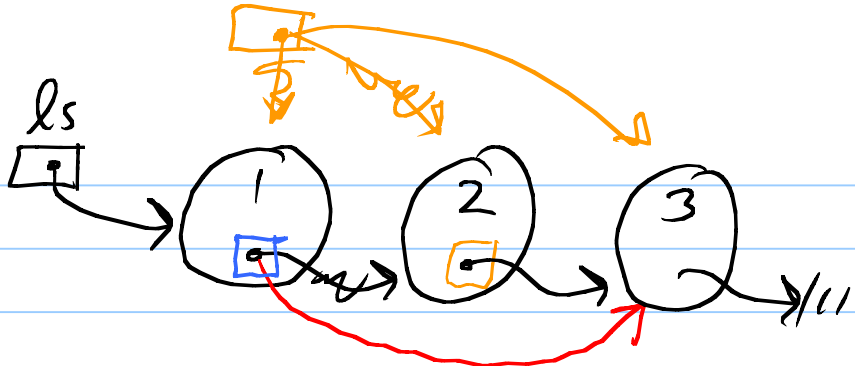
b = a;

b = null;

ListItem d = c.next;



ls.next = ls.next.next;



ListItem a, b;

a = new ListItem(5, null);

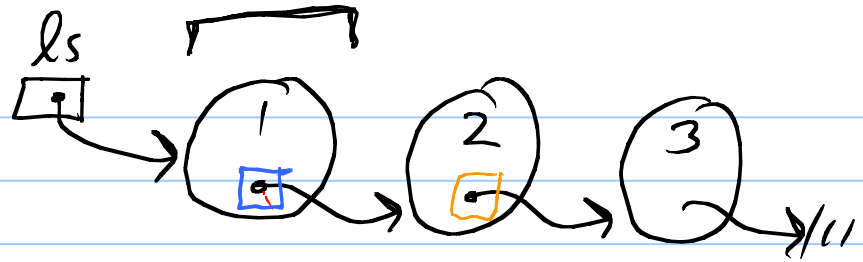
b = new ListItem(7, new ListItem(4, null));

a  
| 100 |

b  
| 104 |

100	5
101	∅
102	4
103	∅
104	7
105	102

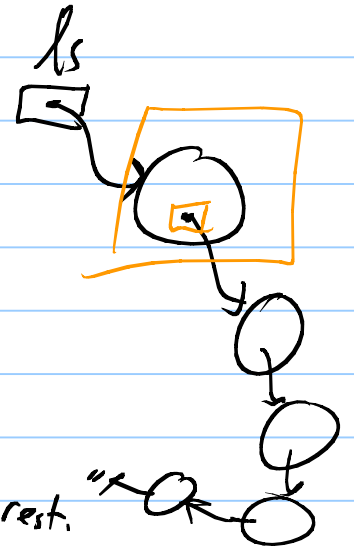
```
class ListItem {
  int value;
  ListItem next;
```



```
public ListItem(int v, ListItem next) {
  value = v;
  this.next = next;
}
```

"1 2 3"

```
public String toString() {
  if (next == null) // no more items
    return value + " ";
  else
    return value + " " + next.toString();
}
```



- Base case: what to do with this item?
- How to combine that with a recursive call on the rest.

```

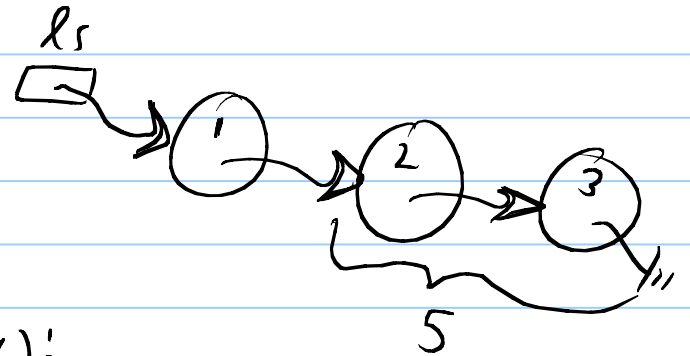
int sum() {
  if (next == null)
    return value;
  else
    return value + next.sum();
}

```

```

int value;
ListItem next;

```



```

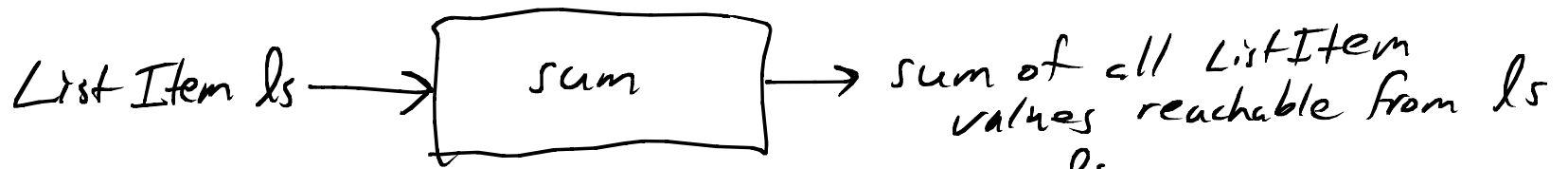
[1 2 3].sum()
1 + [2 3].sum()
1 + 2 + [3].sum()
1 + 2 + 3 ==> 6

```

```

}

```



```
int sum(ListItem ls) {  
    int total = 0;  
    ListItem ptr = ls;  
    while (ptr != null) {  
        total += ptr.value;  
        ptr = ptr.next;  
    }  
    return total;  
}
```

